

PAT-NO: JP408221371A

DOCUMENT-IDENTIFIER: JP 08221371 A

TITLE: SHARED MEMORY CONTROL METHOD AND
CONTROL CIRCUIT FOR THE
METHOD

PUBN-DATE: August 30, 1996

INVENTOR-INFORMATION:

NAME

TAKANO, HIROAKI

ASSIGNEE-INFORMATION:

NAME

OKI ELECTRIC IND CO LTD

COUNTRY

N/A

APPL-NO: JP07027010

APPL-DATE: February 15, 1995

INT-CL (IPC): G06F015/16, G06F012/00 , G11C007/00

ABSTRACT:

PURPOSE: To provide a shared memory control circuit capable of efficiently utilizing a shared memory in the shared memory control circuit for sharing at least one shared memory with plural processing parts requiring a FIFO memory.

CONSTITUTION: This circuit is provided with the shared memory 10 capable of constituting a bi-directional address chain, a write control circuit 40 for writing data to the shared memory 10, a read control circuit 50 for reading the data from the shared memory 10 and a table 20 for memory management and the

table 30 for buffer management to be referred to by the
write control circuit
40 and the read control circuit 50.

COPYRIGHT: (C)1996, JPO

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平8-221371

(43)公開日 平成8年(1996)8月30日

(51)Int.Cl. ^o	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/16	3 5 0		G 0 6 F 15/16	3 5 0 A
	5 9 4		12/00	5 9 4
G 1 1 C 7/00	3 1 8		G 1 1 C 7/00	3 1 8 A

審査請求 未請求 請求項の数7 O L (全 15 頁)

(21)出願番号 特願平7-27010

(22)出願日 平成7年(1995)2月15日

(71)出願人 000000295

沖電気工業株式会社

東京都港区虎ノ門1丁目7番12号

(72)発明者 ▲高▼野 裕昭

東京都港区虎ノ門1丁目7番12号 沖電気
工業株式会社内

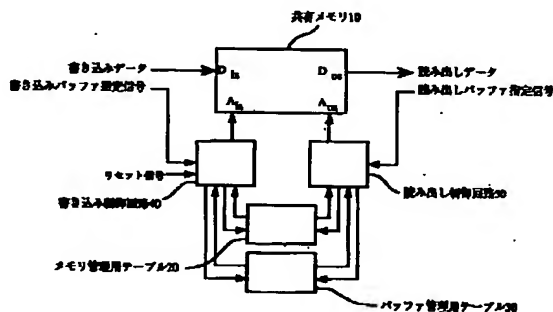
(74)代理人 弁理士 鈴木 敏明

(54)【発明の名称】 共有メモリ制御方法とその制御回路

(57)【要約】

【目的】 F I F Oメモリを必要とする複数の処理部が、少なくとも1つの共有メモリを共有するための共有メモリ制御回路において、共有メモリを効率よく利用できる共有メモリ制御回路を提供することを目的とする。

【構成】 双方向のアドレスチェーンが構成できる共有メモリ10と、共有メモリ10にデータの書き込みを行う書き込み制御回路40と、共有メモリ10からデータの読み出しを行う読み出し制御回路50と、書き込み制御回路40と読み出し制御回路50が参照するためのメモリ管理用テーブル20及びバッファ管理用テーブル30とを有する。



【特許請求の範囲】

【請求項1】 データの書き込み及び読み出しを行う少なくとも第1及び第2の処理部にそれぞれ対応した第1及び第2のFIFOバッファを共有メモリを用いて構成する共有メモリ制御方法において、

前記第1及び第2の処理部のそれぞれに対応した第1及び第2の補償キュー長を設定する第1のステップと、
前記第1の処理部が第1のデータを前記共有メモリに書き込むとき、前記共有メモリに空き領域がある場合は、前記第1のデータを前記空き領域に書き込み、前記共有メモリに空き領域がなく、かつ、前記第1の処理部により前記共有メモリ上に格納されている前記データのキュー長が前記第1の補償キュー長より小さく、かつ、前記第2の処理部により前記共有メモリ上に格納されている前記データのキュー長が前記第2の補償キュー長より大きい場合は、前記第2の処理部により前記共有メモリ上に格納されている第2のデータを廃棄し、前記第2のデータにかえて前記第1のデータを書き込む第2のステップとを有する共有メモリ制御方法。

【請求項2】 データの書き込み及び読み出しを行う少なくとも第1及び第2の処理部にそれぞれ対応した第1及び第2のFIFOバッファを共有メモリを用いて構成する共有メモリ制御方法において、
前記第1及び第2の処理部のそれぞれに対応した第1及び第2の補償キュー長を設定する第1のステップと、
前記第1の処理部が第1のデータを前記共有メモリに書き込むとき、前記共有メモリに空き領域がある場合は、前記第1のデータを前記空き領域に書き込み、前記共有メモリに空き領域がなく、かつ、前記第1の処理部により前記共有メモリ上に格納されている前記データのキュー長が前記第1の補償キュー長より小さく、かつ、前記第2の処理部により前記共有メモリ上に格納されている前記データのキュー長が前記第2の補償キュー長より大きい場合は、前記第2の処理部により前記共有メモリ上に格納されている第2のデータを廃棄し、前記第2のデータにかえて前記第1のデータを書き込む第2のステップとを有する共有メモリ制御方法。

【請求項3】 請求項1または請求項2記載の共有メモリ制御方法において、

さらに、前記第2のステップは、前記共有メモリに空き領域がなく、かつ、前記第1の処理部により前記共有メモリ上に格納されている前記データのキュー長が前記第1の補償キュー長以上の場合は、前記第1のデータを廃棄することを特徴とする共有メモリ制御方法。

【請求項4】 データの書き込み及び読み出しを行う少なくとも第1及び第2の処理部にそれぞれ対応した第1及び第2のFIFOバッファを共有メモリを用いて構成する共有メモリ制御方法において、

前記第1及び第2の処理部のそれぞれに対応した第1及び第2の補償キュー長を設定する第1のステップと、

前記第1の処理部が第1のデータを前記共有メモリに書き込むとき、前記共有メモリに空き領域がある場合は、前記第1のデータを前記空き領域に書き込み、かつ、前記空き領域のアドレスを前記第1のFIFOメモリに格納し、前記共有メモリに空き領域がなく、かつ、前記第1の処理部により前記共有メモリ上に格納されている前記データのキュー長が前記第1の補償キュー長より小さく、かつ、前記第2の処理部により前記共有メモリ上に格納されている前記データのキュー長が前記第2の補償キュー長より大きい場合は、前記第2の処理部により前記共有メモリ上に格納されている第2のデータを廃棄し、前記第2のデータにかえて前記第1のデータを前記共有メモリ上に書き込み、かつ、前記第1のデータが格納されたアドレスを前記第1のFIFOメモリに格納する第2のステップと、

前記第1の処理部が前記共有メモリから前記データを読み出すとき、前記第1のFIFOメモリからFIFOの順序に従って次に読み出されるべき第1のアドレスを読み出し、前記共有メモリの前記第1のアドレスに格納されている前記データを読み出す第3のステップとを有する共有メモリ制御方法。

【請求項5】 データの書き込み及び読み出しを行う少なくとも第1及び第2の処理部にそれぞれ対応した第1及び第2のFIFOバッファを共有メモリを用いて構成する共有メモリ制御方法において、
前記第1及び第2の処理部のそれぞれに対応した第1及び第2の補償キュー長を設定する第1のステップと、
前記第1の処理部が第1のデータを前記共有メモリに書き込むとき、前記共有メモリに空き領域がある場合は、前記第1のデータを前記空き領域に書き込み、かつ、前記空き領域のアドレスを前記第1のFIFOメモリに格納し、前記共有メモリに空き領域がなく、かつ、前記第1の処理部により前記共有メモリ上に格納されている前記データのキュー長が前記第1の補償キュー長より小さく、かつ、前記第2の処理部により前記共有メモリ上に格納されている前記データのキュー長が前記第2の補償キュー長より大きい場合は、前記第2の処理部により前記共有メモリ上に格納されている第2のデータを廃棄し、前記第2のデータにかえて前記第1のデータを前記共有メモリ上に書き込み、かつ、前記第1のデータが格納されたアドレスを前記第1のFIFOメモリに格納する第2のステップと、

前記第1の処理部が前記共有メモリから前記データを読み出すとき、前記第1のFIFOメモリからFIFOの順序に従って次に読み出されるべき第1のアドレスを読み出し、前記共有メモリの前記第1のアドレスに格納されている前記データを読み出す第3のステップとを有する共有メモリ制御方法。

【請求項6】 データの書き込み及び読み出しを行う少なくとも第1及び第2の処理部にそれぞれ対応した第1

キュー長の合計を共有メモリの記憶容量に等しくすると、各論理分割メモリに対して共有メモリを固定的に分割することになり、共有メモリとしての利点が生じない。また、論理分割メモリの補償キュー長の合計を共有メモリの記憶容量を越えるように設定すると、それぞれの補償キュー長で示された各論理分割メモリの記憶容量が補償キュー長まで補償されないことになる。このため、キュー長が補償キュー長に達する以前にデータの書き込みが不可能となる場合が生ずる。

【0005】

【課題を解決するための手段】この発明の共有メモリ制御方法は、データの書き込み及び読み出しを行う少なくとも第1及び第2の処理部にそれぞれ対応した第1及び第2のFIFOバッファを共有メモリを用いて構成する共有メモリ制御方法において、第1及び第2の処理部のそれぞれに対応した第1及び第2の補償キュー長を設定する第1のステップを有し、さらに、第1の処理部が第1のデータを共有メモリに書き込むとき、共有メモリに空き領域がある場合は、第1のデータを空き領域に書き込み、共有メモリに空き領域がなく、かつ、第1の処理部により共有メモリ上に格納されているデータのキュー長が第1の補償キュー長より小さく、かつ、第2の処理部により共有メモリ上に格納されているデータのキュー長が第2の補償キュー長より大きい場合は、第2の処理部により共有メモリ上に格納されている第2のデータを廃棄し、第2のデータにかえて第1のデータを書き込む第2のステップを有するものである。

【0006】

【作用】第1のステップは、第1及び第2の処理部のそれぞれに対応した第1及び第2の補償キュー長を設定するステップである。第2のステップは、第1の処理部が第1のデータを共有メモリに書き込むとき、共有メモリに空き領域がある場合は、第1のデータを空き領域に書き込み、共有メモリに空き領域がなく、かつ、第1の処理部により共有メモリ上に格納されているデータのキュー長が第1の補償キュー長より小さく、かつ、第2の処理部により共有メモリ上に格納されているデータのキュー長が第2の補償キュー長より大きい場合は、第2の処理部により共有メモリ上に格納されている第2のデータを廃棄し、第2のデータにかえて第1のデータを書き込むステップである。

【0007】

【実施例】図1は、この発明の第1の実施例である共有メモリ制御回路を説明するための図である。この第1の実施例の共有メモリ制御回路は、共有メモリ10をn個の論理分割されたFIFO形式のメモリとして用いるための制御回路である。FIFO形式のメモリとは、ファースト・イン・ファースト・アウトの順序でデータが読み出されるのメモリのことである。つまり、データの読み書きを必要とし、しかもFIFO形式のメモリを必要

とする処理部（図示せず）がn個ある場合、これらn個の処理部にメモリを共有させるための制御回路である。ここで、n個の処理部を処理部1、処理部2、…、処理部nと呼ぶことにする。また、共有メモリ10と区別するために、上記処理部に対して論理的に分割してなるFIFO形式のメモリのそれぞれをバッファと呼ぶことにし、処理部 i ($i: 1 \leq i \leq n$, i は整数)に割り当てられたバッファをバッファ i とする。

【0008】共有メモリ10は、書き込みアドレスが入力されるポートAisと、書き込みデータが入力されるポートDisとを有し、さらに、読み出しアドレスが入力されるポートAosと読み出しデータが入力されるポートDosとを有する。書き込み制御回路40及び読み出し制御回路50は、処理部 i に対応したFIFO形式のバッファ i を論理的に構成するための共有メモリ10の制御回路の主要部である。書き込み制御回路40は、書き込みバッファ指定信号により、メモリ管理用テーブル20を参照して書き込みアドレスを求め、上記アドレスを共有メモリ10のポートAisへ送出し、データの書き込みを行う。それと共に、書き込み制御回路40は、アドレスチェーンの再構成に伴う共有メモリ10のデータを更新し、さらに、メモリ管理用テーブル20のデータと、バッファ管理用テーブル30のデータとを更新する。また、書き込み制御回路40は、書き込みアドレスを一時記憶するレジスタを有する。このレジスタをアドレスポインタWPとよぶ。

【0009】読み出し制御回路50は、次に読み出すデータが格納されている共有メモリ10のアドレスを一時記憶するレジスタを有する。このレジスタをアドレスポインタRPとよぶ。読み出しバッファ指定信号によりバッファ i ($i: 1 \leq i \leq n$, i は整数)が指定されると、読み出し制御回路50は、バッファ管理用テーブル30を参照し、アドレスポインタRPに読み出しアドレスを保持して、上記読み出しアドレスを共有メモリ10のポートAosへ送出し、データの読み出しを行う。それと共に、読み出し制御回路50は、アドレスチェーンの再構成に伴う共有メモリ10のデータを更新し、さらに、メモリ管理用テーブル20、バッファ管理用テーブル30を更新する。

【0010】メモリ管理用テーブル20及びバッファ管理用テーブル30は、書き込みアドレスの算出と、アドレスチェーンの再構成に伴う共有メモリ10のデータ更新等のためのテーブルである。

【0011】つぎに、共有メモリ10の論理構成を説明する。第1の実施例の共有メモリ制御回路が、データの書き込み及び読み出しをリスト処理により行うため、共有メモリ10は、図2に示されるデータフォーマットを有する。つまり、共有メモリ10は、図2に示されるメモリセルを論理単位とする。各メモリセルは、書き込みデータを格納する記憶部DATA（以下、単にDATA

と呼ぶ)と、アドレスチェーンを構成するための記憶部QFA(以下、単にQFAと呼ぶ)、記憶部QBA(以下、単にQBAと呼ぶ)、記憶部MFA(以下、単にMFAと呼ぶ)、及び記憶部MBA(以下、単にMBAと呼ぶ)とを有する。

【0012】QFAに格納されるデータは、対応するDATAに格納されているデータが属するバッファ i ($1 \leq i \leq n$, i は整数)を論理構成するアドレスチェーンの前方向のアドレスである。つまり、DATAに格納されているデータが属するバッファ i のデータの中で、当該DATAに格納されているデータより1データ先に格納されたデータが格納されたメモリセルのアドレスである。以後、バッファ i のデータで構成されているアドレスチェーンをバッファ用アドレスチェーン i ($i: 1 \leq i \leq n$, i は整数)と呼ぶことにする。アドレスチェーンの詳細については後述する。

【0013】QBAに格納されるデータは、対応するDATAに格納されているデータが属するバッファ i のデータで構成されるアドレスチェーンの後方方向のアドレスである。つまり、対応するDATAに格納されているデータが属するバッファ i のデータの中で、当該DATAに格納されているデータより1データ後に格納されたデータが格納されたメモリセルのアドレスである。

【0014】MFAに格納されるデータは、メモリ管理用アドレスチェーンの前方向のアドレスである。MBAに格納されるデータは、メモリ管理用アドレスチェーンの後方方向のアドレスである。メモリ管理用アドレスチェーンの詳細については後述する。

【0015】つぎに、この実施例において構成されるアドレスチェーンについて説明する。まず、バッファ用アドレスチェーンについて説明する。図3は、バッファ用アドレスチェーン i をQFA及びQBAを用いて構成していることを示す図である。図中、メモリセル080は、アドレス080から始まるメモリセルであり、メモリセル080のDATAには、バッファ i に書き込まれたデータ(m)が格納されている。ここで、データ(m)は、バッファ i に書き込まれたデータの中で、現在 m 番目に読み出されるべきデータである。メモリセル080のQFAには、バッファ i に書き込まれたデータの中で現在($m-1$)番目に読み出されるべきデータ($m-1$)、つまり、データ(m)の1データ前に読みだされるべきデータが格納されているメモリセルのアドレスが格納されている。また、メモリセル080のQBAには、バッファ i に書き込まれたデータの中で現在($m+1$)番目に読み出されるべきデータ($m+1$)、つまり、データ m の1データ後に読み出されるデータが格納されているメモリセルのアドレスが格納されている。図3は、バッファ i に書き込まれたデータの順序に従ってメモリセルを並べた概念図であり、左から右に行くほど遅く書き込まれたデータが格納されているメモリ

セルとなる。また、矢印は、矢印の起点となる格納領域に矢印の終点で示されるアドレスが格納されていることを示している。このように、QFA及びQBAにより、双方向のバッファ用アドレスチェーンが構成されている。このように双方向の1本のアドレスチェーンにより、結合されるメモリセルの書き込み時期の順序関係がわかる。

【0016】つぎに、メモリ管理用テーブル20及びバッファ管理用テーブル30を表1及び表2を用いて説明する。まず、メモリ管理用テーブル20を表1を用いて説明する。メモリ管理用テーブル20は、記憶部A1(以下、単にA1と呼ぶ)、記憶部B1(以下、単にB1と呼ぶ)、記憶部C1(以下、単にC1と呼ぶ)とを有する。A1には、共有メモリ10においてデータが入っていない空のメモリセルの集合を、アドレスチェーンによって論理結合した場合の先頭のメモリセルのアドレスが格納されている。このアドレスチェーンを空きメモリセル用アドレスチェーンと呼ぶことにする。また、B1には、空きメモリセル用アドレスチェーンによるメモリセルの論理結合において、最後尾のメモリセルのアドレスが格納されている。さらに、C1には、空きメモリセル用アドレスチェーンのキュー長が格納されている。ここで、空きメモリセル用アドレスチェーンのキュー長とは、空きメモリセル用アドレスチェーンにより結合されたメモリセル数を示している。この空メモリ用アドレスチェーンは、後述する共有メモリ10の初期化のときにつくられる。

【0017】

【表1】

A1	空メモリセル用アドレスチェーン ヘッドアドレス
B1	空メモリセル用アドレスチェーン テイルアドレス
C1	空メモリセル用アドレスチェーン キュー長
A2	補償キュー長超過メモリセル用アドレスチェーン ヘッドアドレス
B2	補償キュー長超過メモリセル用アドレスチェーン テイルアドレス
C2	補償キュー長超過メモリセル用アドレスチェーン キュー長

【0018】

【表2】

Di	バッファ用アドレスチェーンi ヘッドアドレス
Ei	バッファ用アドレスチェーンi テイルアドレス
Fi	バッファiのキュー長
Gi	バッファiの補償キュー長
Hi	バッファ用アドレスチェーンiに結合された(補償 キュー長+1)番目のメモリセルのアドレス

【0019】メモリ管理用テーブル20は、さらに記憶部A2（以下、単にA2と呼ぶ）、記憶部B2（以下、単にB2と呼ぶ）及び記憶部C2（以下、単にC2と呼ぶ）とを有する。A2には、補償キュー長超過メモリセル用アドレスチェーンによる論理結合における先頭のメモリセルのアドレスが格納されている。B2には、補償キュー長超過メモリセル用アドレスチェーンによるメモリセルの論理結合における最後尾のメモリセルのアドレスが格納されている。C2には、補償キュー長超過メモリセル用アドレスチェーンのキュー長が格納されている。

【0020】ここで、補償キュー長超過メモリセル用アドレスチェーンとは、バッファごとに予め決められたバッファ記憶容量を越えて書き込まれたデータを結合したものである。つまり、バッファが記憶しているデータ量が、すでに補償キュー長を越えており、その状況下でバッファにデータが書き込まれた場合、当該データが格納されたメモリセルが、論理的に補償キュー長超過メモリセル用アドレスチェーンに結合されていく。したがって、補償キュー長超過メモリセル用アドレスチェーンにより結合されているメモリセルに格納されているデータには、例えばバッファ1に属するものがあつたり、バッファ5に属するものがあつたりする。初期状態においては、この補償キュー長超過メモリセル用アドレスチェーンは構成されていない。

【0021】つぎに、バッファ管理用テーブル30を表2を用いて説明する。バッファ管理用テーブル30には、バッファごとに結合されたバッファ用アドレスチェーンに関するデータが格納されている。バッファ管理用テーブル30は、記憶部Di（ $i: 1 \leq i \leq n$ 、 i は整数）（以下、単にDiと呼ぶ）、記憶部Ei（ $i: 1 \leq i \leq n$ 、 i は整数）（以下、単にEiと呼ぶ）、記憶部Fi（ $i: 1 \leq i \leq n$ 、 i は整数）（以下、単にfiと呼ぶ）、記憶部Gi（ $i: 1 \leq i \leq n$ 、 i は整数）（以下、単にGiと呼ぶ）、記憶部Hi（ $i: 1 \leq i \leq n$ 、 i は整数）（以下、単にHiと呼ぶ）とを有する。バッファ用アドレスチェーンiに関するデータとして、Diにヘッドアドレスが格納され、Eiにテイルアドレスが格納され、Fiにキュー長が格納され、Giに補償キュー長が格納され、Hiに（補償キュー長+1）番目のアドレスが格納されている。

【0022】ヘッドアドレスとは、アドレスチェーンによるメモリセルの論理結合における先頭のメモリセルのアドレスである。テイルアドレスとは、アドレスチェーンによるメモリセルの論理結合における最後尾のメモリセルのアドレスである。キュー長とは、バッファ用アドレスチェーンiのキュー長である。補償キュー長とは、バッファiに対して予め設定された記憶容量である。この補償キュー長により、バッファiの記憶容量が補償される。したがって、 $G1 + G2 + \dots + Gi + \dots + Gn$

（ $i: 1 \leq i \leq n$ 、 i は整数）の値を共有メモリ10の記憶容量と等しくなるように設定することが、共有メモリ10の有効利用の点で望ましい。もちろん、 $G1 + G2 + \dots + Gi + \dots + Gn$ の値が共有メモリ10の記憶容量より小さくなるように設定しても構わない。（補償キュー長+1）番目のアドレスとは、補償キュー長超過メモリセル用アドレスチェーンに結合されたバッファiに属するデータの中で、最初に読み出されるべきデータが格納されたメモリセルのアドレスである。

【0023】それでは、この発明の第1の実施例の共有メモリ制御回路の動作を説明する。まず、図1における共有メモリ10の初期化について説明する。図4は、共有メモリ10の初期化を説明するためのフローチャートである。図1の書き込み制御回路40にリセット信号が入力されると、書き込み制御回路40は、リスト処理により共有メモリ10のメモリセルを論理的に接続する（手続き（11）、（12））。つまり、メモリセルのMFA及びMBAを使用して、アドレスチェーンを構成する。このアドレスチェーンで結合されたメモリセルのデータ格納部は空であるので、初期化により生成されるアドレスチェーンは、上説した空きメモリセル用アドレスチェーンである。さらに、空きメモリセル用アドレスチェーンにより結合されたメモリセルに基づいて、メモリ管理用テーブル20のA1にヘッドアドレスを、B1にテイルアドレスを、C1にキュー長を書き込む（手続き（13））。以上の動作により、共有メモリ10の初期化が完了する。

【0024】つぎに、書き込み動作を説明する。書き込みデータが、図1の共有メモリ10のポートDisに入力されると、それとともに書き込みバッファをバッファiと指定する書き込みバッファ指定信号が、書き込み制御回路40に入力される。書き込みデータがバッファを指定するデータをヘッダとしてもつ場合は、書き込み制御回路40で上記書き込みデータのヘッダを解説してもよい。

【0025】図5は、データの書き込み動作を示すフローチャートである。バッファiを指定する書き込みバッファ指定信号が入力されると、図1の書き込み制御回路40は、メモリ管理用テーブル20の空きメモリセル用アドレスチェーンのキュー長C1の値が、ゼロかどうかを判定する（手続き（21）、（22））。C1の値が

11

ゼロでないときは、空きメモリセルが存在しているので、図5中のモジュール1を実行する(手続き(24))。

【0026】図6にモジュール1のフローチャートを示す。まず、アドレスポインタWPの値を、図1中のメモリ管理用テーブル20の空きメモリセル用アドレスチェーンのヘッドアドレスA1の値に更新する(手続き(31))。つぎに、メモリ管理用テーブル20の空きメモリセル用アドレスチェーンのヘッドアドレスA1の値を、アドレスA1のメモリセルのMBAの値に更新する(手続き(32))。つぎに、アドレスポインタWPの値を図1の共有メモリ10のポートAisに送出し、共有メモリ10のアドレスWPに書き込みデータを書き込む(手続き(33))。つぎに、メモリ管理用テーブル20の空きメモリセル用アドレスチェーンのキュー長C1の値を1つデクリメントする(手続き(34))。以上の動作により、空きメモリセル用アドレスチェーンが更新された。

【0027】つぎに、アドレスWPのメモリセルのQFAの値を、図1中のバッファ管理用テーブル30のバッファ用アドレスチェーンiのテイルアドレスEiの値に更新する(手続き(35))。アドレスWPのメモリセルのQFAに示されているアドレス、このアドレスのメモリセルのQBAの値を、WPの値に更新する(手続き(36))。つぎに、バッファ管理用テーブル30のバッファ用アドレスチェーンiのキュー長Fiの値を1つインクリメントする(手続き(37))。以上により、バッファ用アドレスチェーンiが更新された。

【0028】つぎに、バッファ管理用テーブル30のバッファ用アドレスチェーンiのキュー長Fiの値と、バッファiの補償キュー長Giとを比較し、 $Fi \leq Gi$ であれば、モジュール1を終了する。 $Fi > Gi$ であれば、以下に示す補償キュー長超過メモリセル用アドレスチェーンの再構成を行う(手続き(38))。まず、 $Fi = Gi + 1$ のときのみ、つまり補償キュー長超過メモリセル用アドレスチェーンにバッファiのデータが1つも結合されていないとき、バッファ管理用テーブル30の(補償キュー長+1)番目のアドレスH1の値を、WPの値に更新する(手続き(39)、(40))。つぎに、入力データが補償キュー長超過メモリセル用アドレスチェーンに結合されるので、メモリ管理用テーブル20の補償キュー長超過メモリセル用アドレスチェーンのキュー長C2の値を1つインクリメントする(手続き(41))。つぎに、アドレスWPのメモリセルのMFAの値を、メモリ管理用テーブル20の補償キュー長超過メモリセル用アドレスチェーンのテイルアドレスB2の値に更新する(手続き(42))。つぎに、アドレスB2のメモリセルのMBAの値を、WPの値に更新する(手続き(43))。つぎに、メモリ管理用テーブル20の補償キュー長超過メモリセル用アドレスチェーンの

12

テイルアドレスB2の値を、WPの値に更新する(手続き(44))。モジュール1の手続きは以上の通りである。

【0029】つぎに、図5のフローチャートにおいて、 $C1 = 0$ の場合のフローを説明する。まず、 $Fi \geq Gi$ の場合、つまりバッファ用アドレスチェーンiのキュー長Fiの値が、バッファiの補償キュー長Giの値以上大きいとき、入力データを廃棄し終了する(手続き(26))。言い換えれば、空のメモリセルがなく、しかも図1の共有メモリ10に格納されているバッファiのデータ量が、すでに補償キュー長を越えている場合は、入力データが廃棄されることになる。一方、 $Fi < Gi$ の場合は、モジュール2を実行する(手続き(25))。

【0030】図7にモジュール2のフローチャートを示す。まず、アドレスポインタWPの値を、補償キュー長超過メモリセル用アドレスチェーンのヘッドアドレスA2の値に更新する(手続き(51))。つぎに、メモリ管理用テーブル20の補償キュー長超過メモリセル用アドレスチェーンのヘッドアドレスA2の値を、アドレスA2のメモリセルのMBAの値に更新する(手続き(52))。つぎに、アドレスポインタWPの値を図1の共有メモリ10のポートAisに送出し、入力データの書き込みを行う(手続き(53))。以上の手続き(51)、(52)、(53)により、アドレスWPのセルが、補償キュー長超過メモリセル用アドレスチェーンからリリースされ、廃棄されたことになる。つぎに、メモリ管理用テーブル20の補償キュー長超過メモリセル用アドレスチェーンのキュー長C2の値を、1つデクリメントする(手続き(54))。以上の動作により、補償キュー長超過メモリセル用アドレスチェーンの先頭のメモリセルが、補償キュー長超過メモリセル用アドレスチェーンからリリースされる。これにより、補償キュー長超過メモリセル用アドレスチェーンが更新された。つぎに、アドレスWPのメモリセルのQFAに示されているアドレス、このアドレスのメモリセルのQBAの値を、アドレスWPのメモリセルのQBAの値に更新する(手続き(55))。つぎに、アドレスWPのメモリセルのQBAに示されているアドレス、このアドレスのメモリセルのQFAの値を、アドレスWPのメモリセルのQFAの値に更新する(手続き(56))。この手続き(55)、(56)動作により、バッファ用アドレスチェーンh($h: 1 \leq h \leq n, h \neq i$)により、廃棄されたデータと接続されているメモリセル同士が接続されたことになる。これにより、バッファ用アドレスチェーンh($h: 1 \leq h \leq n, h \neq i$)が再構成された。

【0031】つぎに、アドレスWPのメモリセルのQFAの値を、バッファ管理用テーブルのバッファ用アドレスチェーンiのテイルアドレスEiの値に更新する(手続き(57))。つぎに、アドレスEiのメモリセルのQBAの値を、WPの値に更新する(手続き(5

8))。

【0032】つぎに、バッファ管理用テーブルのバッファ用アドレスチェーン*i*のテイルアドレス*Ei*の値を、アドレスポインタWPの値に更新する(手続き(59))。つぎに、バッファ管理用テーブル30のバッファ用アドレスチェーン*i*のキュー長*Fi*の値を、1つインクリメントする(手続き(60))。以上により、モジュール2の手続きが完了し、書き込み動作が終了する。

【0033】つぎに、図8を用いて読み出し動作を説明する。読み出すバッファをバッファ*i*と指定する読み出しバッファ指定信号が、図1中の読み出し制御回路50に入力されると、読み出し制御回路50は、アドレスポインタを算出し、さらにメモリ管理用テーブル20、バッファ管理用テーブル30及び共有メモリ10のメモリセルのデータの更新を行う。図8は、データの読み出し動作を示すフローチャートである。バッファ*i*に対するデータの出力要求がくると、図1の読み出し制御回路40は、バッファ管理用テーブル30のバッファ*i*のキュー長*Fi*がゼロかどうかを判定する(手続き(80)、(81))。 *Fi*の値がゼロのときは、バッファ*i*にデータが存在していないので、データの読み出しは行われない。*Fi*の値がゼロでないときは、以下の手続きを行う。まず、アドレスポインタRPの値を、バッファ管理用テーブル30のバッファ*i*のヘッドアドレス*Di*の値に更新する(手続き(82))。つぎに、バッファ管理用テーブル30のバッファ*i*のヘッドアドレス*Di*の値を、アドレス*Di*のメモリセルのQBAの値に更新する(手続き(83))。これにより、アドレスRPのメモリセルが、バッファ用アドレスチェーン*i*からリリースされたことになる。

【0034】つぎに、バッファ管理用テーブル30のバッファ*i*のキュー長*Fi*の値が、バッファ*i*の補償キュー長*Gi*の値より大きいときは、以下の2つの手続きを行う(手続き(84))。まず、アドレスHiのメモリセルのMFAに示されるアドレス、このアドレスのメモリセルのMBAの値を、アドレスHiのメモリセルのMBAの値に更新する(手続き(85))。さらに、アドレスHiのメモリセルのMBAに示されるアドレス、このアドレスのメモリセルのMFAの値を、アドレスHiのメモリセルのMFAの値に更新する(手続き(86))。以上の手続き(85)、(86)により、補償キュー長超過メモリセル用アドレスチェーンで結合されたバッファ*i*のデータの中で、最初に読み出されるデータが格納されたメモリセルが、補償キュー長超過メモリセル用アドレスチェーンからリリースされる。さらに、補償キュー長超過メモリセル用アドレスチェーンによって、リリースされたメモリセルと結合されたメモリセル同士が結合される。以上により、補償キュー長超過メモリセル用アドレスチェーンが再構成されたことになる。

【0035】つぎに、図1の読み出し制御回路50は、アドレスポインタRPの値を共有メモリ10のポートDisに送出する(手続き(87))。これにより、アドレスRPのメモリセルのデータが読み出される。バッファ*i*からデータが1つ読み出されたので、バッファ監視用テーブル30のバッファ*i*のキュー長*Fi*の値を1つデクリメントする(手続き(88))。つぎに、バッファ*i*のキュー長*Fi*の値が、バッファ*i*の補償キュー長*Gi*より大きいとき、つまりデータを読み出した後、以前としてバッファ*i*のデータ量が補償キュー長を越えているときは、バッファ監視用テーブル30のバッファ*i*の(補償キュー長+1)番目のアドレスHiの値を、アドレスHiのメモリセルのQBAの値に更新する(手続き(89)、(90))。

【0036】つぎに、空きメモリセル用アドレスチェーンを再構成する。まず、アドレスBiのメモリセルのMBAの値を、アドレスポインタRPの値に更新する(手続き(91))。さらに、アドレスRPのメモリセルのMFAの値を、メモリ管理用テーブルの空きメモリセル用アドレスチェーンのテイルアドレスB1の値に更新する(手続き(92))。以上の手続きにより、空のメモリセルとなったアドレスRPのメモリセルが、空きメモリセル用アドレスチェーンの最後尾に結合された。

【0037】つぎに、メモリ管理用テーブル20の空きメモリセル用アドレスチェーンのテイルアドレスB1の値を、RPの値に更新する(手続き(93))。さらに、空きメモリセル用アドレスチェーンのキュー長C1の値を1つインクリメントして、読み出しのための手続きが終了する(手続き(94))。

【0038】この第1の実施例では、空きメモリセルを空きメモリセル用アドレスチェーンで結合し、さらに空きメモリセル用アドレスチェーンのヘッドアドレスをヘッドアドレスA1に格納した。これは、読み出し速度を高めるための手法であり、必ずしも空きメモリセルをアドレスチェーンにより結合する必要はない。なぜなら、空きメモリセルには、順序関係が存在しないからである。空きメモリセルをアドレスチェーンにより結合しない場合は、メモリセルが空きであることを示す第1のフラグをたて、入力データを書き込むときに、第1のフラグがたっているメモリセルのうち、任意の1つのメモリセルを選べばよい。

【0039】また、第1の実施例では、補償キュー長超過メモリセルを補償キュー長超過メモリセル用アドレスチェーンで結合し、さらに補償キュー長超過メモリセル用アドレスチェーンのヘッドアドレスをヘッドアドレスA1に格納した。これもまた、読み出し速度を高めるための手法であり、必ずしも、補償キュー長超過空きメモリセル同士を属するバッファに関係なくアドレスチェーンにより結合する必要はない。補償キュー長超過メモリセルをアドレスチェーンにより結合しない場合は、メモ

リセルに格納されているデータが補償キュー長を超過してバッファに書き込まれていることを示す第3のフラグをたて、入力データをバッファ*i*書き込む際に、バッファ*h*のセルを廃棄する場合は、第3のフラグがたっているメモリセルのうち、任意の1つのメモリセルを選べばよい。

【0040】以上説明したように、空きメモリセル用アドレスチェーン及び補償キュー長超過メモリセル用アドレスチェーンを構成しない場合は、メモリセルが空きであることを示す第1のフラグ、メモリセルに格納されているデータが補償キュー長を超過してバッファに書き込まれていることを示す第2のフラグ及びメモリセルに格納されているデータが補償キュー長内でバッファに書き込まれていることを示す第3のフラグを示す格納部をメモリセルに用意する必要がある。また、この場合、メモリセル内の記憶部MFA、MBA及びメモリ管理用テーブル20は必要ない。さらに、読み出し動作及び書き込む動作において、空きメモリセル用アドレスチェーン及び補償キュー長超過メモリセル用アドレスチェーンの再構成も必要ない。この構成によれば、使用する制御用のメモリ領域が、第1の実施例の構成より少なくて済むという効果が生じる。

【0041】つぎに、この発明の第2の実施例を示す。図9は、この発明の第2の実施例を説明するための図である。第2の実施例において、第1の実施例と大きく異なる点は、第2の実施例の共有メモリ110が、メモリセルによる論理構成をとってはならず、単に、アドレスを指定してデータを書き込み、またアドレスを指定してデータを読み出す構成のメモリである点である。また、次に大きく異なる点は、処理部*i*によって書き込まれたデータ間の順序関係をアドレスチェーンによって関係付けるのではなく、処理部ごとに用意されたアドレス用FIFOメモリにアドレスを格納していくことにより、その順序の関係付けを行う点である。

【0042】それでは、第2の実施例の構成を図9を用いて説明する。この第2の実施例の共有メモリ制御回路は、第1の実施例と同様、共有メモリ110を*n*個の論理分割されたFIFO形式のメモリとして用いるための制御回路である。つまり、データの読み書きを必要とし、しかもFIFO形式のメモリを必要とする処理部（図示せず）が*n*個ある場合、これら*n*個の処理部にメモリを共有させるための制御回路である。ここで、第1の実施例と同様に*n*個の処理部を処理部1、処理部2、…、処理部*n*と呼ぶことにする。また、共有メモリ110と区別するために、上記処理部に対して論理的に分割してなるFIFO形式のメモリのそれぞれをバッファと呼ぶことにし、処理部*i*（ $1 \leq i \leq n$ 、*i*は整数）に割り当てられたバッファをバッファ*i*とする。

【0043】共有メモリ110は、書き込みアドレスが入力されるポートAisと、書き込みデータが入力され

るポートDisとを有し、さらに、読み出しアドレスが入力されるポートAosと読み出しデータが入力されるポートDosとを有する。書き込み制御回路140及び読み出し制御回路150は、処理部*i*に対応したFIFO形式のバッファ*i*を論理的に構成するための共有メモリ110の制御回路の主要部である。

【0044】書き込み制御回路140は、空きメモリアドレス用メモリ170及び補償キュー長値格納メモリ180と接続されており、それぞれのメモリに格納された情報を読み出す。また、書き込み制御回路140は、空きメモリアドレス用メモリ170及び補償キュー長値格納メモリ180からの情報並びに書き込みバッファ指定信号に基づいて書き込みアドレスを求め、上記書き込みアドレスを共有メモリ110のポートAisへ送出し、書き込みデータの書き込みを行う。したがって、書き込み制御回路140は、書き込みアドレスを一時的に保持する書き込みアドレスレジスタを有する。さらに、書き込み制御回路140は、処理部ごとに設けられたアドレス用FIFOメモリ161～16*n*に接続されており、書き込みバッファ指定信号に基づいて、アドレス用FIFOメモリ161～16*n*の1つを選択し、そのアドレス用FIFOメモリに書き込みアドレス値を格納する。

【0045】補償キュー長値格納メモリ180は、バッファごとに予め設定された補償キュー長を格納するメモリである。ここで、補償キュー長の合計を共有メモリ110に格納できるキュー長の総数に等しくすることが、共有メモリ110を有効利用する点で望ましい。

【0046】読み出し制御回路150は、処理部ごと、つまりバッファごとに設けられたアドレス用FIFOメモリ161～16*n*に接続されており、書き込みバッファ指定信号に基づいて、アドレス用FIFOメモリ161～16*n*の1つを選択し、そのアドレス用FIFOメモリから読み出しアドレス値を読み出す。また、読み出し制御回路150は、読み出しアドレスを一時的に保持する読み出しアドレスレジスタを有する。

【0047】つぎに、第2の実施例の共有メモリ制御回路における書き込み動作を説明する。書き込みデータが、図9の共有メモリ110のポートDisに入力されると、それとともに書き込みバッファをバッファ*i*と指定する書き込みバッファ指定信号が、書き込み制御回路140に入力される。書き込みデータがバッファを指定するデータをヘッダとしてもつ場合は、書き込み制御回路140で上記書き込みデータのヘッダを解読してもよい。

【0048】図10は、データの書き込み動作を説明するためのフローチャートである。以下に、書き込みデータの書き込み動作を説明する。書き込みデータが、図9の共有メモリ110のポートDisに入力されると、それとともに書き込みバッファをバッファ*i*と指定する書き込みバッファ指定信号が、書き込み制御回路140に

10

20

30

40

50

17

入力される。書き込みデータがバッファを指定するデータをヘッダとしてもつ場合は、書き込み制御回路140が上記書き込みデータのヘッダを解読してもよい。

【0049】バッファ*i*を指定する書き込みバッファ指定信号が、書き込み制御回路140に入力されると、書き込み制御回路140は、共有メモリ110の現在使用されていないメモリのアドレスを空きメモリアドレス用メモリ170から読み出す。空きメモリアドレス用メモリ170は、FIFO形式のメモリであり、現在使用されていないメモリのアドレスをすべて記憶している。この第2の実施例では空きメモリアドレス用メモリ170としてFIFO形式のメモリを使用しているが、空きメモリアドレス用メモリ170は、FIFO形式のメモリに限らず、FIFO形式のメモリまたはランダムに読み出す形式のメモリでもよい。つまり、空きメモリアドレス用メモリ170は、共有メモリ110の空きメモリアドレスの任意の1つを読み出せることができるメモリであれば足りる。

【0050】空きメモリアドレス用メモリ170に空きメモリアドレスが格納されている場合、書き込み制御回路140は、その1つを書き込みアドレスレジスタに保持し、図10のモジュール11を次に実行する。また、空きメモリアドレス用メモリ170に空きメモリアドレスが格納されていない場合、書き込み制御回路140は、図10の手続き(123)を実行する(手続き(121)、(122))。

【0051】つぎに、空きメモリアドレス用メモリ170に空きメモリアドレスが格納されている場合に実行される図10のモジュール11の手続きを説明する。図11は、モジュール11の手続きを説明するためのフローチャートである。まず、書き込み制御回路140は、書き込みアドレスレジスタが保持しているアドレス値を共有メモリ110のポートAisへ送出し、書き込みデータを共有メモリ110に書き込む(手続き(131))。つぎに、書き込み制御回路140は、バッファ*i*を指定する書き込みバッファ指定信号により、アドレス用FIFOメモリの中でアドレス用FIFOメモリ16*i*を特定する。その後、書き込み制御回路140は、書き込みアドレスレジスタが保持しているアドレス値をアドレス用FIFOメモリ16*i*に格納し、モジュール11の手続きを終了する(手続き(132))。

【0052】つぎに、空きメモリアドレス用メモリ170に空きメモリアドレスが格納されていない場合に実行される図10の手続き(123)を以下に説明する。まず、書き込み制御回路140は、補償キュー長値格納メモリ180に格納されたバッファごとの補償キュー長のデータの中で、バッファ*i*に対応した補償キュー長のデータを読み出し、その補償キュー長の値とアドレス用FIFOメモリ16*i*に格納されているアドレスデータのキュー長の値とを比較する。その結果、アドレス用FI

18

FOメモリ16*i*に格納されているアドレスデータのキュー長の値が、補償キュー長の値より小さい場合は、図10のモジュール12を実行する。また、アドレス用FIFOメモリ16*i*に格納されているアドレスデータのキュー長の値が、補償キュー長の値より小さくない場合は、入力される書き込みデータを廃棄する。

【0053】以下に、アドレス用FIFOメモリ16*i*に格納されているアドレスデータのキュー長の値が補償キュー長の値より小さい場合に、実行される図10のモジュール12の手続きを説明する。図12は、モジュール12の手続きを説明するためのフローチャートである。まず、書き込み制御回路140は、補償キュー長値格納メモリ180に格納される補償キュー長の値とアドレス用FIFOメモリ16*k* ($k: 1 \leq k \leq n$)のキュー長とをそれぞれ比較する(手続き(141))。そして、格納されているアドレスデータのキュー長の値が補償キュー長の値より大きいアドレス用FIFOメモリを特定する(手続き(142))。特定されたアドレス用FIFOメモリの中で、その1つを選び、当該アドレス用FIFOメモリに格納されているアドレスデータの1つを廃棄する(手続き(143))。ここで、特定されたアドレス用FIFOメモリの中でその1つを選ぶ最適な方法の1つとしては、補償キュー長の値からアドレスデータのキュー長の値までの超過分が最も大きいアドレス用FIFOメモリを選ぶ方法があげられる。

【0054】書き込み制御回路140は、上述の廃棄されたアドレスデータの値を書き込みアドレスレジスタに保持し、そのアドレス値を共有メモリ110のポートAisへ送出する。これにより、書き込みデータを共有メモリ110へ書き込む(手続き(144))。さらに、書き込み制御回路140は、書き込みアドレスレジスタが保持しているアドレス値をアドレス用FIFOメモリ16*i*に格納し(手続き(145))、モジュール12の手続きを終了する。

【0055】つぎに、第2の実施例の共有メモリ制御回路における読み出し動作を説明する。読み出しバッファをバッファ*i*と指定する読み出しバッファ指定信号が、読み出し制御回路150に入力されると、読み出し制御回路150は、アドレス用FIFOメモリ16*i*からFIFOの順序関係により読み出されたアドレスデータを読み出し、当該アドレスデータを読み出しアドレスレジスタに保持する。その後、読み出し制御回路150は、当該アドレスデータを共有メモリ110のポートAosへ送出し、共有メモリ110の当該アドレスに格納されているデータを読み出す。

【0056】また、読み出し制御回路150の読み出しアドレスメモリへ保持されたアドレスデータは、空きメモリアドレス用メモリ170に格納される。

【0057】上説した第2の実施例は、処理部ごとのFIFOメモリを共有メモリ110を用いて実現するため

の共有メモリ制御回路であるが、この第2の実施例は、さらに、処理部ごとのFIFOメモリを共有メモリ110を用いて実現するための共有メモリ制御回路にも変形可能である。この場合、図9のアドレス用FIFOメモリ161~16nをFIFO形式のメモリに置換すれば、容易に実現可能である。

【0058】以上詳細に説明したように、この発明の第2の実施例の共有メモリ制御回路は、処理部ごとにアドレスチェーンを構成する方法をとってはならず、それぞれのバッファの書き込みデータの論理的順序関係をアドレス用FIFOメモリを用いることによって実現している。

【0059】一般に、FIFO形式のメモリは、連続的なメモリ領域にデータを格納していく。つまり、書き込みに関しては、書き込みアドレスを示すライトポインタを書き込みごとにインクリメントすることにより実現される。また、読み出しに関しては、読み出しアドレスを示すリードポインタを読み出しごとにインクリメントすることにより実現される。このように、ライトポインタおよびリードポインタを用いることにより、ファースト・イン・ファースト・アウトの順序関係をつくる。したがって、第2の実施例の共有メモリ制御回路において、例えば、図9のアドレス用FIFOメモリが部分的なメモリエラーを起こし、アドレスデータが1部破壊したとしても、アドレス用FIFOメモリが連続的なメモリ領域にデータを格納しているため、破壊したアドレスの次のアドレスを特定でき、破壊したアドレスデータを除いて、アドレスデータ間のファースト・イン・ファースト・アウトの順序関係が保たれる。つまり、共有メモリ110上の読み出しデータ間のファースト・イン・ファースト・アウトの順序関係は、アドレスデータの1部破壊によっては、破壊したアドレスデータに対応する読み出しデータを除いては崩れることはない。

【0060】

【発明の効果】以上詳細に説明したようにこの発明によれば、各処理部は、処理部毎に予め設定された補償キュー

一長に達するまでは、共有メモリを論理的に分割したバッファへの書き込みが補償される。さらに、共有メモリ内に空きメモリが存在する場合は、たとえ処理部が書き込んだデータ量が予め設定された補償キュー長を越えても、データの書き込みができる。したがって、共有メモリにおける未使用領域の有効利用が図れる。

【図面の簡単な説明】

【図1】第1の実施例を示すブロック図である。

【図2】共有メモリ10のデータフォーマットを示す図である。

【図3】アドレスチェーンを説明するための図である。

【図4】初期化を説明するフローチャートである。

【図5】第1の実施例の書き込み動作を説明するフローチャートである。

【図6】モジュール1の手続きを説明するフローチャートである。

【図7】モジュール2の手続きを説明するフローチャートである。

【図8】第1の実施例の読み込み動作を説明するフローチャートである。

【図9】第2の実施例を示すブロック図である。

【図10】第1の実施例の書き込み動作を説明するフローチャートである。

【図11】モジュール11の手続きを説明するフローチャートである。

【図12】モジュール12の手続きを説明するフローチャートである。

【符号の説明】

10、110 共有メモリ

20 メモリ管理用テーブル

30 バッファ管理用テーブル

40、140 書き込み制御回路

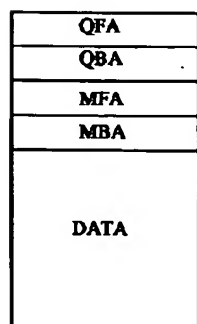
50、150 読み込み制御回路

161~16n アドレス用FIFOメモリ

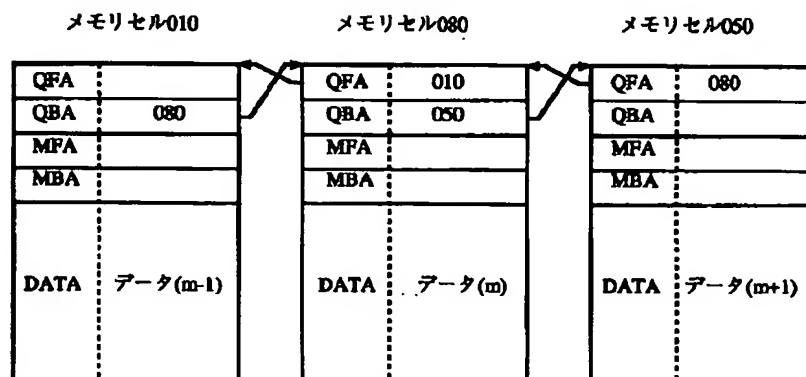
170 空きメモリアドレス用メモリ

180 補償キュー長値格納メモリ

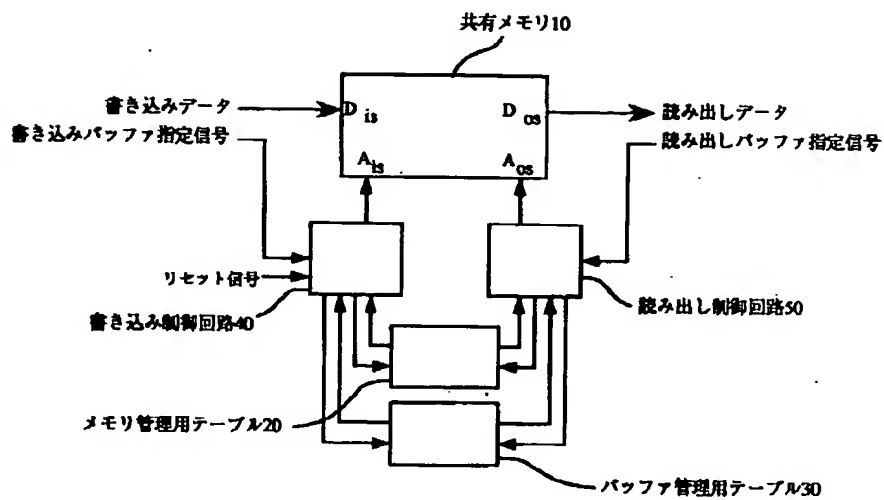
【図2】



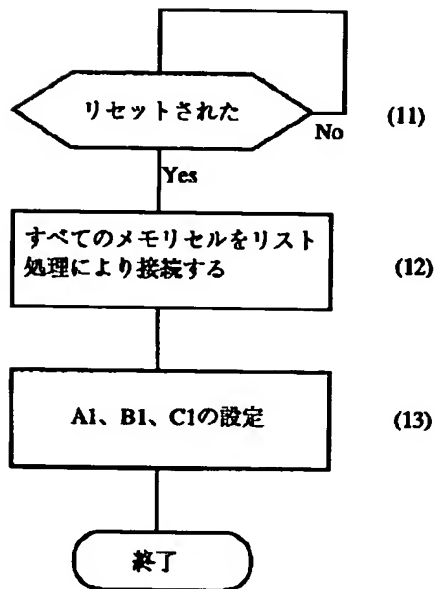
【図3】



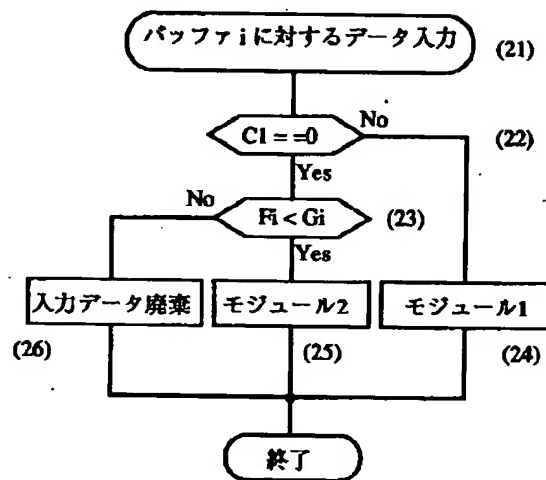
【図1】



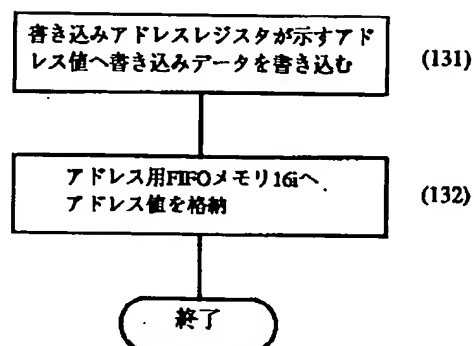
【図4】



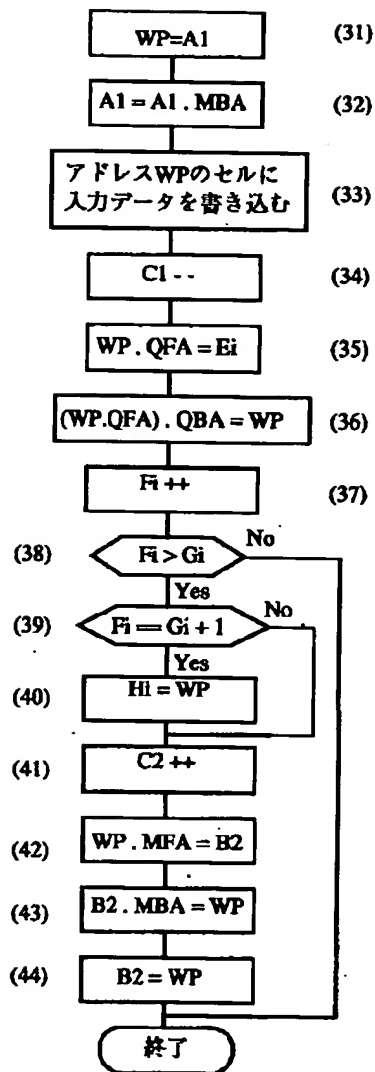
【図5】



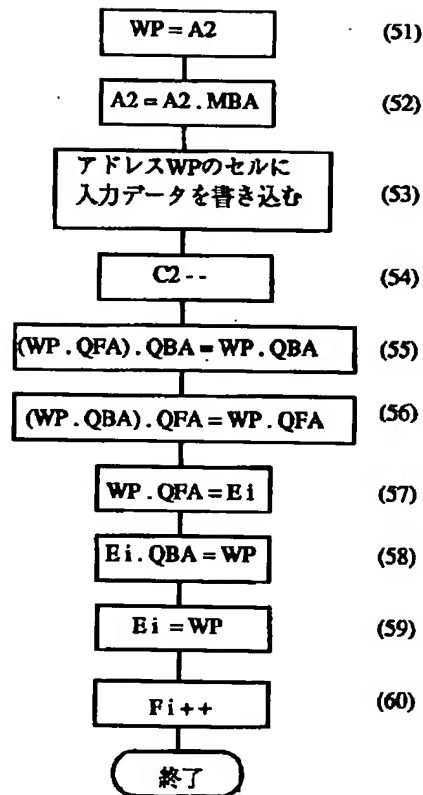
【図11】



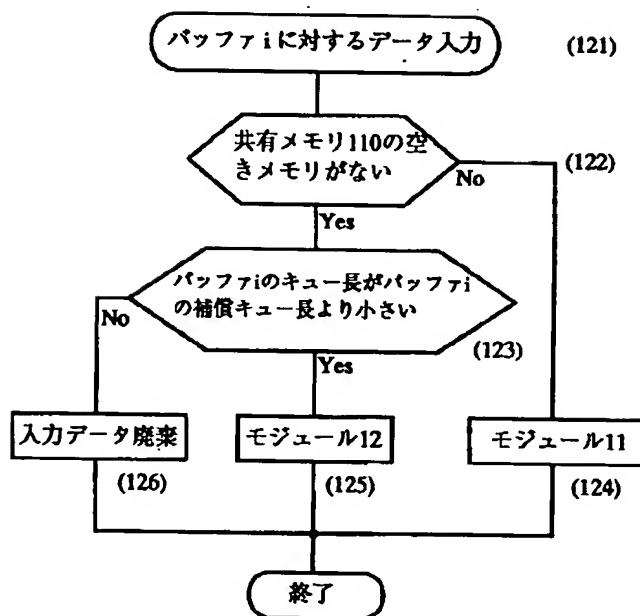
【図6】



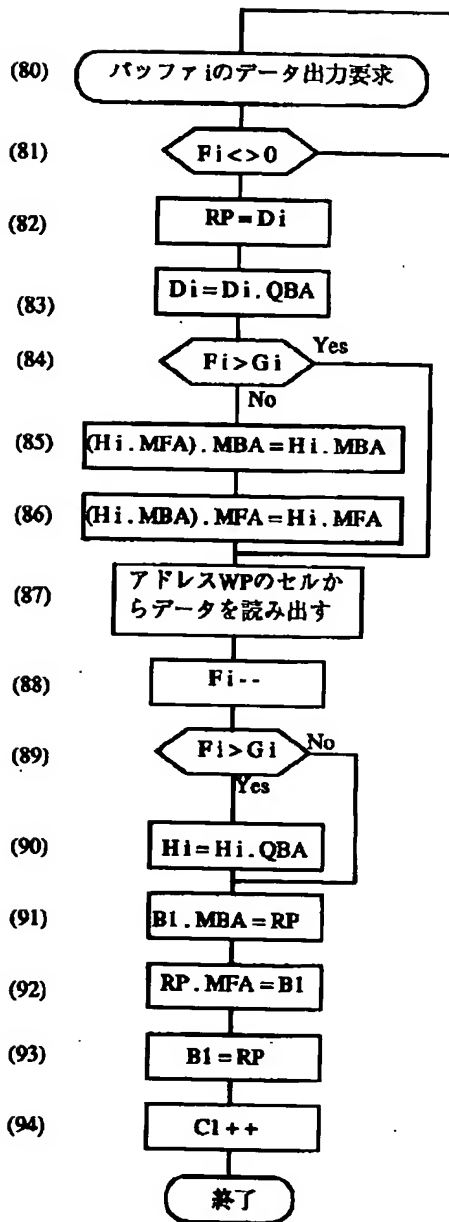
【図7】



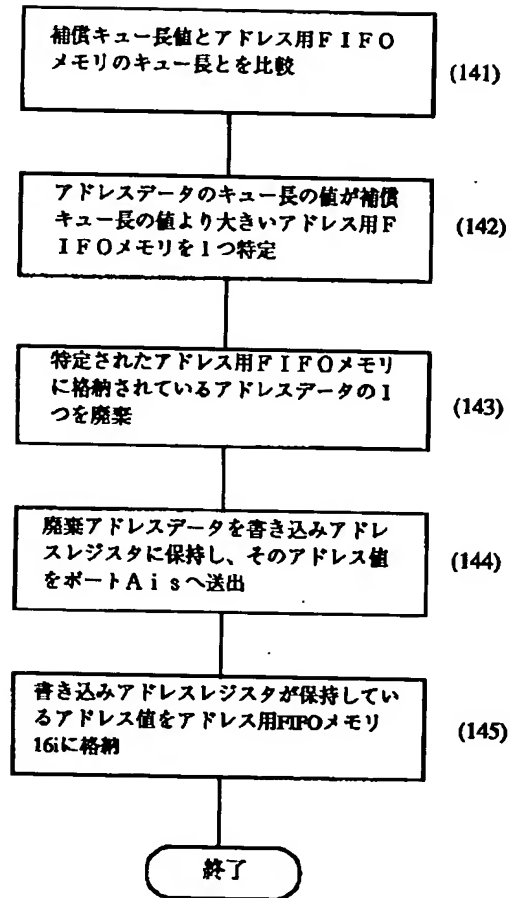
【図10】



【図8】



【図12】



【図9】

